ANSWERS

U5.5 Tuning

BugBotLab

Feedback control · University · about 30 min

What this lesson is about

Finding the gain where it oscillates, and working back from there to something usable.

- The Ziegler-Nichols method
- Finding Ku on this robot
- Tuning by hand, with the chart
- Do it in the right order
- The gains are not universal

Questions

7 marks in all

1. The Ziegler-Nichols PID row from the lesson, for a measured Ku and Tu. What does it print? [1 mark]

```
KU, TU = 10.0, 1.6
kp = 0.6 * KU
ki = 1.2 * KU / TU
kd = 0.075 * KU * TU
print(round(kp, 2), round(ki, 2), round(kd, 2))
```

Answer:

```
6.0 7.5 1.2
```

Kp = 0.6 x 10 = 6, Ki = 1.2 x 10 / 1.6 = 7.5, Kd = 0.075 x 10 x 1.6 = 1.2.

2. Ku = 8 and Tu = 2 s. What Ki does the Ziegler-Nichols PI row give? Give two decimal places. [1 mark]

Answer: 2.16 (accept within 0.005). The PI row gives Ki = 0.54 Ku / Tu = 0.54 x 8 / 2 = 2.16.

3. Ku = 9. What Kp does the Ziegler-Nichols P-only row give? [1 mark]

Answer: 4.5 (accept within 0.001). P only uses Kp = 0.5 Ku = 4.5.

4. Ziegler-Nichols gains make the robot swing lively, each swing a quarter of the last. What is a common and sensible next move? [1 mark]

- ☐ **A** Halve Kp
- ☐ **B** Double Ki
- ☐ **C** Remove Kd
- ☐ **D** Double Ku and recompute

Answer: A. Ziegler-Nichols aims for quarter amplitude decay, which most people find too aggressive. It is a starting point, and halving Kp is the usual first adjustment. On a big step, such as the lesson's quarter turn, the Ki it gives usually needs cutting as well.

5. The response arrives, settles short of the target, and stays short. What should change? [1 mark]

- ☐ **A** Add or raise Ki
- ☐ **B** Raise Kd
- ☐ **C** Lower Kp
- ☐ **D** Filter the measurement

> **Answer:** A. A steady offset is what the integral term exists to remove.

6. The response shows a slow, wallowing oscillation. What should change? [1 mark]

- ☐ **A** Lower Ki
- ☐ **B** Raise Kp
- ☐ **C** Raise Kd
- ☐ **D** Run the loop slower

> **Answer:** A. Slow, large swings are the look of too much integral action. Fast oscillation at the loop rate points at Kp instead.

7. In what order does the lesson say to bring in the three terms? [1 mark]

- ☐ **A** P, then D, then I
- ☐ **B** P, then I, then D
- ☐ **C** I, then P, then D
- ☐ **D** All three at once from Ziegler-Nichols, then fine tune

> **Answer:** A. Get P fast without ringing, use D to remove the ring, and add I last. Adding I early makes the offset you are trying to judge keep disappearing.

The task: find the ultimate gain

Run the heading loop at several gains, print the swing at each, and print `ku:` and `tu:`.

```
from bugbot import *
connect()

DT = 0.25
```

The hint students can ask for: Run the same loop several times, each at a higher gain, and watch the error. Steady oscillation that neither grows nor dies is the ultimate gain. The period is the time between two peaks.

A solution

```
from bugbot import *
connect()

DT = 0.25

def swing(kp, ticks=32):
    """Turn a quarter turn at this gain; report how much the error swings, and how fast, in the second half."""
    target = heading() + 90
    errors = []
    for tick in range(ticks):
        error = (target - heading() + 180) % 360 - 180
        errors.append(error)
        plot("kp %s" % kp, error)
        drive(0, 0, max(-100.0, min(100.0, kp * error)))
        wait(DT)
    stop()
    wait(0.5)
    late = errors[ticks // 2:]
    mean = sum(late) / len(late)
    crossings = 0
    for a, b in zip(late, late[1:]):
        if (a - mean) * (b - mean) < 0:
            crossings += 1
    span = max(late) - min(late)
    period = len(late) * DT * 2.0 / crossings if crossings else 0.0
    return span, period

ku, tu = 0.0, 0.0
for kp in (2.0, 4.0, 6.0, 8.0, 10.0):
    span, period = swing(kp)
    print("kp", kp, "swing", round(span, 1), "period", round(period, 2))
    if span > 10.0 and ku == 0.0:
        ku, tu = kp, period
print("ku:", ku)
print("tu:", round(tu, 2))
```

Any program that meets the task's checks is marked correct in the simulator; this is one way, not the only way.